

1. 以下哪一个语言的发明者没有获得图灵奖（ ）

A. C B. C++ C. Pascal D. Smalltalk

2. 在8位二进制补码中，10101010表示的数是十进制下的：（ ）

A. 176 B. -86 C. -85 D. -84

3. 以下哪一台计算机是被实际制造出来的（ ）

A. 巴贝奇差分机II号 B. 巴贝奇分析机 C. EDVAC D. 图灵机

4. 链接存储的存储结构所占存储空间（ ）。

A. 分两部分，一部分存放结点值，另一部分存放表示结点间关系的指针

B. 只有一部分，存放结点值

C. 只有一部分，存放表示结点间关系的指针

D. 分两部分，一部分存放结点值，另一部分存放结点所占单元数

5. 若 $f[1]=1, f[2]=5/6, 3f[n+1]=5f[n]-2f[n-1]$ ，则随着i的增大， $f[i]$ 将接近于（ ）

A. $1/3$ B. $\sqrt{2}-1$ C. $1/2$ D. 1

6. 有一富翁，为了确保自己的人身安全，雇了双胞胎兄弟两个作保镖。兄弟两个确实尽职尽责，为了保证主人的安全，他们做出如下行事准则：

 - a. 每周一、二、三，哥哥说谎；
 - b. 每逢四、五、六，弟弟说谎；
 - c. 其他时间两人都说真话。

一天，富翁的一个朋友急着找富翁，他知道要想找到富翁只能问兄弟俩，并且他也知道兄弟俩个的做事准则，但不知道谁是哥哥，谁是弟弟。另外，如果要知道答案，就必须知道今天是星期几。于是他便问其中的一个人：昨天是谁说的日子？结果两人都说：是我说谎的日子。你能猜出今天是星期几吗？（ ）

A. 星期三 B. 星期日 C. 星期六 D. 星期四

7. 高度为n的均衡的二叉树是指，如果去掉叶结点及相应的树枝，它应该是高度为n-1的满二叉树。在这里，树高等于叶结点的最大深度，根结点的深度为0，如果某个均衡的二叉树共有2381个结点，则该树的树高为（ ）

A. 10 B. 11 C. 12 D. 13

8. 关键字序列(8, 9, 10, 4, 5, 6, 20, 1, 2)只能是下列排序算法中()的两趟排序后的结果。

A. 选择排序 B. 冒泡排序 C. 插入排序 D. 快速排序

9. 公共汽车起点站于每小时的10分,30分,55分发车,该顾客不知发车时间,在每小时内的任一时刻随机到达车站,求乘客候车时间的数学期望(准确到秒). ()

A. 8分40秒 B. 10分25秒 C. 22分30秒 D. 35分30秒

10. 最大容量为n的循环队列，队尾指针是rear，队头是front，则队空的条件是（ ）。

A. $(\text{rear}+1)\%n == \text{front}$ B. $\text{rear} == \text{front}$

C. $\text{rear} + 1 == \text{front}$ D. $(\text{rear}-1)\%n == \text{front}.$

11. 在含n个顶点和e条边的无向图的邻接矩阵中，零元素的个数为（ ）

A. $(n-1)^2-e$ B. $2e$ C. n^2-e D. n^2-2e

12. 某文本包含240个汉字、39个数字以及666个字母，若将其强制转换为一个char数组，则数组的长度为（ ）
- A. 945 B. 279 C. 1851 D. 1185
13. 若一棵二叉树具有10个度为2的结点，5个度为1的结点，则度为0的结点的个数是（ ）。
- A. 9 B. 11 C. 15 D. 不能确定
14. 中缀表达式 $A-(B+C/D)*E$ 的后缀表达式是（ ）
- A. $AB-C+D/E*$ B. $ABC+D/-E*$ C. $ABCD/E*+-$ D. $ABCD/+E*-$
15. 二维数组A的每个元素是由10个字符组成的串，其行下标 $i=0,1,...,8$ ，列下标 $j=1,2,...,10$ 。若A按行先存储，元素A[8][5]的起始地址与当A按列先存储时的元素（ ）的起始地址相同。设每个字符占一个字节。
- A. A[8][5] B. A[3][10] C. A[5][8] D. A[0][9]
16. 以下IPv4地址可以表示C类网络主机的是：（ ）
- A. 115.236.49.57 B. 210.33.19.103 C. 172.16.1.55 D. 10.196.1.100
17. 若让元素1, 2, 3, 4, 5依次进栈，则出栈次序不可能出现（ ）的情况。
- A. 5,4,3,2,1
B. 2,1,5,4,3
C. 4,3,1,2,5
D. 1,2,5,4,3
18. 以下事项不违反CSP-J/S考场纪律的是（ ）。
- A. 为了看时间将电话手表带入考场。
B. 在开考2小时内离场
C. 在开始考试前在机器上默写模版代码
D. 向监考员举手示意需要去洗手间
19. 下列说法正确的是（ ） 算法
- A. SPFA算法无法用来判断给定图是否存在负环
B. 当图中不存在负权环但是存在负权边，Dijkstra算法一定不能求最短路
C. 当图中不存在负权环但是存在负权边，bellman-ford算法一定能求最短路
D. 相比于稀疏图，在稠密图上更适合使用SPFA算法
20. 前序遍历序列与中序遍历序列相同的二叉树为（ ）
- A. 根结点无左子树的二叉树
B. 根结点无右子树的二叉树
C. 只有根结点的二叉树或非叶子结点只有左子树的二叉树
D. 只有根结点的二叉树或非叶子结点只有右子树的二叉树

二、单项选择题（共2题，每题5分,每空标号处均填选择的答案对应的大写字母）

21. 在一个AOE网中，从源点到汇点的路径不止一条，每条路径都可以用来刻画一项工程的具体实施过程，当路径上所有活动都完成了，整个工程才算完成。关键活动，指的是不按期完成就会影响整个工程完成的活动。请问以下哪个选项不是关键活动（ ）。
- a1活动: 1节点指向2, 长度为6;
a2活动: 1节点指向3, 长度为4;
a3活动: 1节点指向4, 长度为5;
a4活动: 2节点指向5, 长度为1;
a5活动: 3节点指向5, 长度为1;
a6活动: 4节点指向6, 长度为2;
a7活动: 5节点指向7, 长度为7;

a8活动:5节点指向8,长度为5;
a9活动:6节点指向8,长度为4;
a10活动:7节点指向9,长度为2;
a11活动:8节点指向9,长度为4.

A.a1

B.a5

C.a7

D.a8

22. 你一定玩过汉诺塔游戏:有三根柱子,在一根柱子上有若干盘子,保证自上而下从小到大堆放,现要将所有盘子移动

到最后一根柱子上,保证:1、移动过程中,始终保证小盘子在大盘子上;2、在三根柱子间一次只能移动一个盘子。现在

规则稍有一些变化(相信难不倒聪明的你)——四根柱子,其他规则不变,问把A柱子上20个盘子移动到D上的最小步数(

)?

A.289

B.2023

C.524287

D.1048575

三、单项选择题(共4题,每题8分,每空标号处均填选择的答案对应的大写字母)

1. 有以下程序:

```
#include<bits/stdc++.h>
using namespace std;
int m[101][101];
int main() {
    int a,i,k;
    cin>>a;
    int c=a*a;
    i=1,k=(a+1)/2;
    for(int j=1; j<=c; j++) {
        m[i][k]=j;
        if(j%a==0)
            if(i==a) i=1;
            else i++;
        else {
            if(i==1)i=a;
            else i--;
            if(k==a)k=1;
            else k++;
        }
    }
    for(int j=1; j<=a; j++) cout<<m[3][j]<<" ";
}
```

输入:

5

程序运行后的输出结果是()。

A.4 7 15 18 6

B. 3 7 15 19 22

C.4 6 13 20 22

D. 23 5 7 14 16

2. 阅读程序写结果,

```

#include<bits/stdc++.h>
using namespace std;
int map1[55][55][55];
int vis[55][55][55];
typedef struct {
int z,x,y;
} node;
int aa[6][3]= {1,0,0, -1,0,0, 0,0,1, 0,1,0, 0,0,-1, 0,-1,0};
int a,b,c,tt,yy;
void fun() {
    node w,d;
    w.z=1;
    w.x=1;
    w.y=1;
    queue <node > q;
    q.push(w);
    while(!q.empty()) {
        d=q.front();
        for(int i=0; i<6; i++) {
            w.z=d.z+aa[i][0];
            w.x=d.x+aa[i][1];
            w.y=d.y+aa[i][2];
            if(w.z>=1&&w.z<=a&&w.x>=1&&w.x<=b&&w.y>=1&&w.y<=c &&map1[w.z][w.x]
[w.y]==0&&vis[w.z][w.x][w.y]==0) {
                vis[w.z][w.x][w.y]=vis[d.z][d.x][d.y]+1;
                if(vis[w.z][w.x][w.y]>tt)
                    return ;
                if(w.z==a&&w.x==b&&w.y==c) {
                    yy=1;
                    printf("%d\n",vis[d.z][d.x][d.y]);
                    return ;
                } else q.push(w);
            }
        }
        q.pop();
    }
    return ;
}
int main () {
int i,j,k;
scanf("%d %d %d %d",&a,&b,&c,&tt);
memset(vis,0,sizeof(vis));
for(i=1; i<=a; i++)
    for(j=1; j<=b; j++)
        for(k=1; k<=c; k++)
            scanf("%d",&map1[i][j][k]);

yy=0;
vis[1][1][1]=1;
fun();
if(yy==0) printf("-1\n");
return 0;
}

```

输入:

```

3 3 4 20
0 1 1 1
0 0 1 1
0 1 1 1
1 1 1 1
1 0 0 1
0 1 1 1
0 0 0 0
0 1 1 0
0 1 1 0

```

输出 () :

A.11 B.15 C.17 D.23

3.

```

#include<iostream>
#include<cstdio>
using namespace std;
#define ll long long
ll g(ll k){
    if(k<=1) return k;
    return (2002*g(k-1)+2003*g(k-2))%2005;
}
int main(){
    ll n;
    cin>>n;
    cout<<g(n);
    return 0;
}

```

输入

2005

输出 ()

A.31 B.2004 C.29 D.37

4.

```

#include <bits/stdc++.h>
using namespace std;
const int maxNum = 1005;
int shift[maxNum];
int Sd(const string& T, const string& P) {
    int n = T.length();
    int m = P.length();
    for(int i = 0; i < maxNum; i++) {
        shift[i] = m + 1;
    }
    for(int i = 0; i < m; i++) {
        shift[P[i]] = m - i;
    }
    int s = 0;
    int j;
    while(s <= n - m) {

```

```

j = 0;
while(T[s + j] == P[j]) {
    j++;
    if(j >= m) {
        return s;
    }
}
s += shift[T[s + m]];
}
return -1;
}
int main() {
    string T, P;
    getline(cin, T);
    getline(cin, P);
    int res = Sd(T, P);
    cout << res << endl;
    return 0;
}

```

输入:

```

adasfasfasfasgaegagfasdf
gf

```

输出: ()

A.4 B.7 C.13 D.18

四、程序填空（每空2分，共28分,每空标号处均填选择的答案对应的大写字母）

给出n件物品的价值和体积，问在总体积不超过v时的第k大价值。

输入格式:

第一行为三个整数n,v,k，第二行n个整数表示这n件物品的价值，第三行n个整数表示这n件物品的体积
 $N \leq 100, v \leq 1000, k \leq 30$ ，价值 ≤ 1000 ，体积 ≤ 1000

输出格式:

输出总体积不超过v的第k大价值

样例输入:

```

5 10 2
1 2 3 4 5
5 4 3 2 1

```

样例输出:

```

12
#include<stdio.h>
#include<string.h>
int t,n,v,k,dp[1001][35],value[101],cost[101];
void Kth() {
    int t1[35],t2[35],res1,res2,res;
    memset(dp,0,sizeof(dp)); //初始化
    for(int i=1; i<=n; i++) //01背包
        for(int j=v; j>=cost[i]; j--) {
            for(int l=1; l<=k; l++) {
                t1[l]=__27__+value[i];
                t2[l]=__28__;
            }
        }
    }
}

```

```

    }
    res1=res2=res=___29___;
    t1[k+1]=t2[k+1]=-1;
    while(res<=k&&(res1<=k||res2<=k)) {
        if(___30___)
            dp[j][res]=t1[res1++];
        else
            dp[j][res]=t2[res2++];
        if(___31___!=___32___)
            res++;
    }
}
}
int main() {
    scanf("%d%d%d",&n,&v,&k);
    for(int i=1; i<=n; i++)
        scanf("%d",&value[i]);
    for(int i=1; i<=n; i++)
        scanf("%d",&cost[i]);
    kth();
    printf("%d",___33___);
    return 0;
}

```

27. A. dp[v-cost[i]][1] B. dp[j-cost[i]][1] C. dp[v-cost[1]][i] D. dp[j-cost[1]][i]

28. A. cost[i] B. cost[j] C. dp[j][1] D. dp[i][1]

29. A. -1 B. 0 C. 1 D. 1001

30. A. res1<res2 B. res1>res2 C. t1[res1]<t2[res2] D. t1[res1]>t2[res2]

31. A. dp[j][res-1] B. dp[i][res-1] C. dp[i][res+1] D. dp[j][res+1]

32. A. dp[i][res] B. dp[j][res] C. dp[1][res] D. dp[1][res+1]

33. A. dp[v][k-1] B. dp[n][k-1] C. dp[v][k] D. dp[n][k]

N个人正在排队。队列中任意两个人A和B，如果他们是相邻或他们之间没有人比A或B高，那么他们是可以互相看得见的。现要求有多少对人可以互相看见。

输入N和这N个人的身高，输出s对人。

```

#include<bits/stdc++.h>
using namespace std;
const int maxn=5e5+5;
int s[maxn],w[maxn],tot,n,tmp;
long long ans;
int main(){
    scanf("%d",&n);
    for(int i=1;i<=n;i++){
        scanf("%d",&tmp);
        if(tmp<s[___34___]) ans+=1LL,s[___35___]=tmp,w[tot]=1;
        else{
            int width=0;
            while(___36___&&tmp>s[tot]) ans+=1LL*w[tot],___37___;
            while(tot&&___38___) ans+=1LL*w[tot],width+=___39___, tot--;
            if(tot) ans+=1LL;
            s[++tot]=tmp;
            w[___40___]=width+1;
        }
    }
    printf("%lld",ans);
    return 0;
}

```

```
}  
34. A. tot      B.maxn      C.tmp      D.s[tmp]  
35. A.tot+1    B.++tot      C.tot++    D.tot  
36. A.tot+1    B.++tot      C.tot++    D.tot  
37. A.tot+1    B.tot--      C.tot++    D.tot  
38. A.tmp<s[tot] B.tmp!=s[tot] C.tmp==s[tot] D.tmp>s[tot]  
39. A.w[tot]    B.w[tmp]     C.s[tot]   D.s[tmp]  
40. A.tot+1    B.++tot      C.tot++    D.tot
```